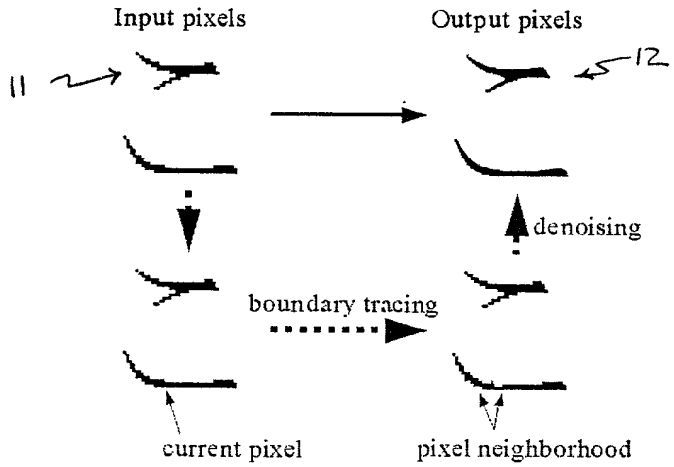

Input pixels
Output pixels
11
12
denoising
boundary tracing
current pixel
pixel neighborhood

FIG. 1

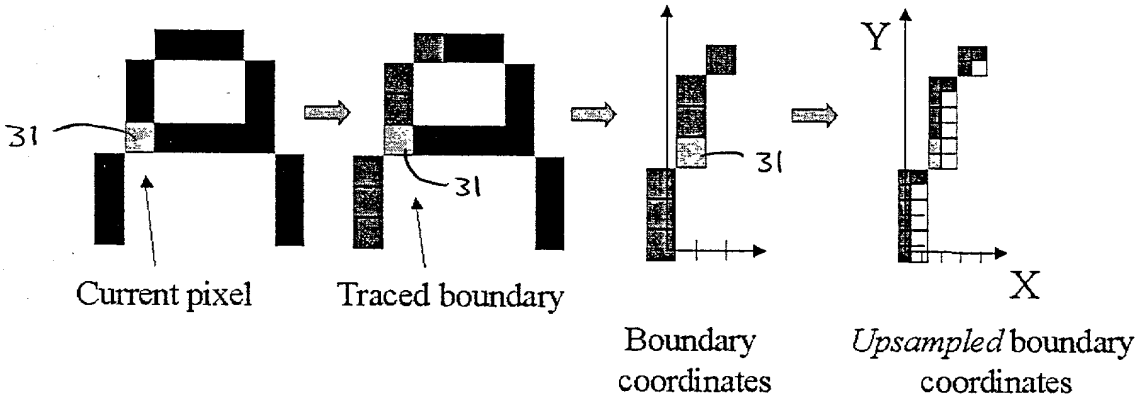

31
31
Y
31
X
Current pixel
Traced boundary
Boundary coordinates
Upsampled boundary coordinates

FIG. 3

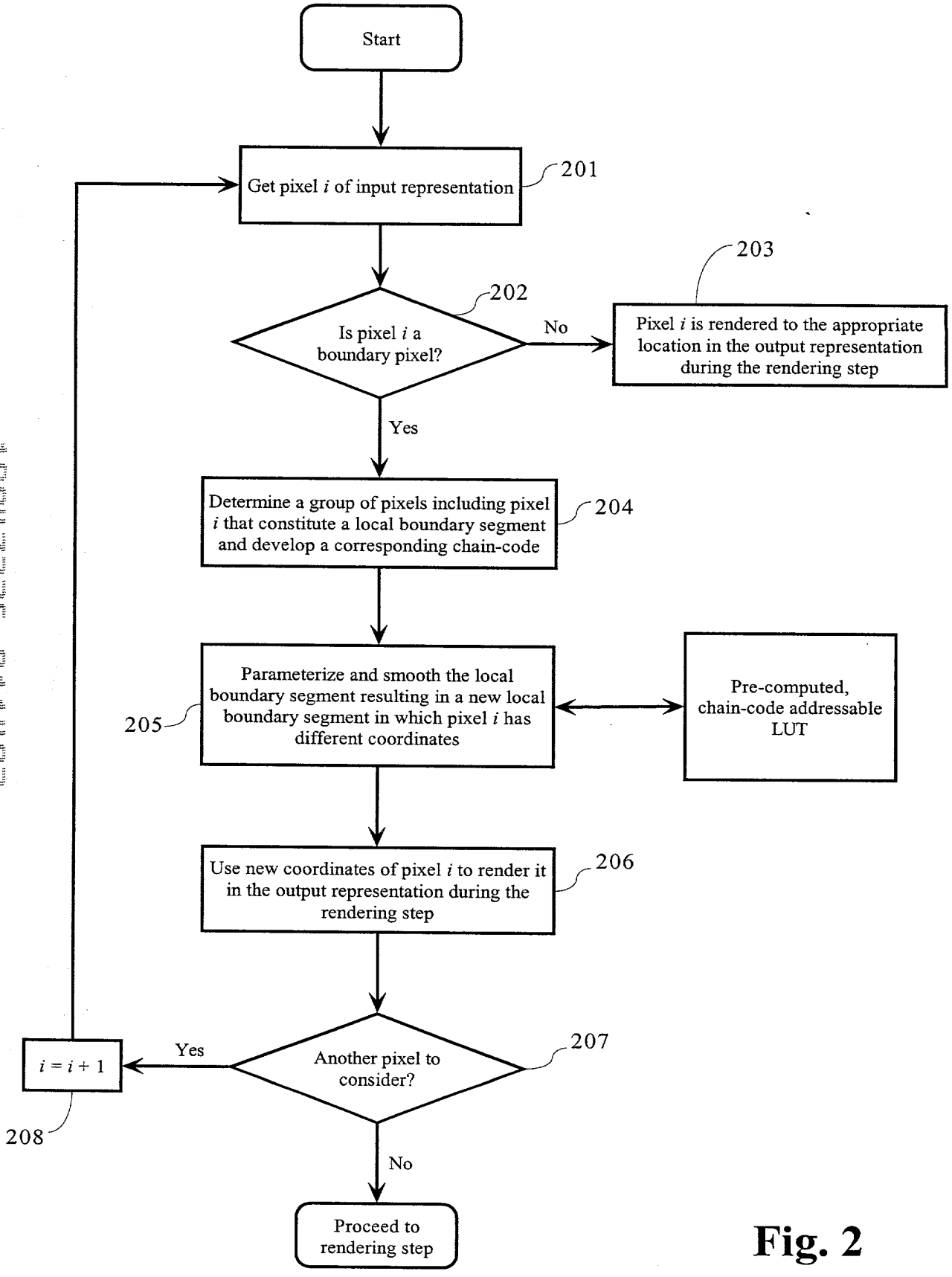

Start
Get pixel i of input representation
201
Is pixel i a boundary pixel?
202
No
Pixel i is rendered to the appropriate location in the output representation during the rendering step
203
Yes
Determine a group of pixels including pixel i that constitute a local boundary segment and develop a corresponding chain-code
204
Parameterize and smooth the local boundary segment resulting in a new local boundary segment in which pixel i has different coordinates
205
Pre-computed, chain-code addressable LUT
Use new coordinates of pixel i to render it in the output representation during the rendering step
206
Another pixel to consider?
207
Yes
i = i + 1
208
No
Proceed to rendering step

Fig. 2

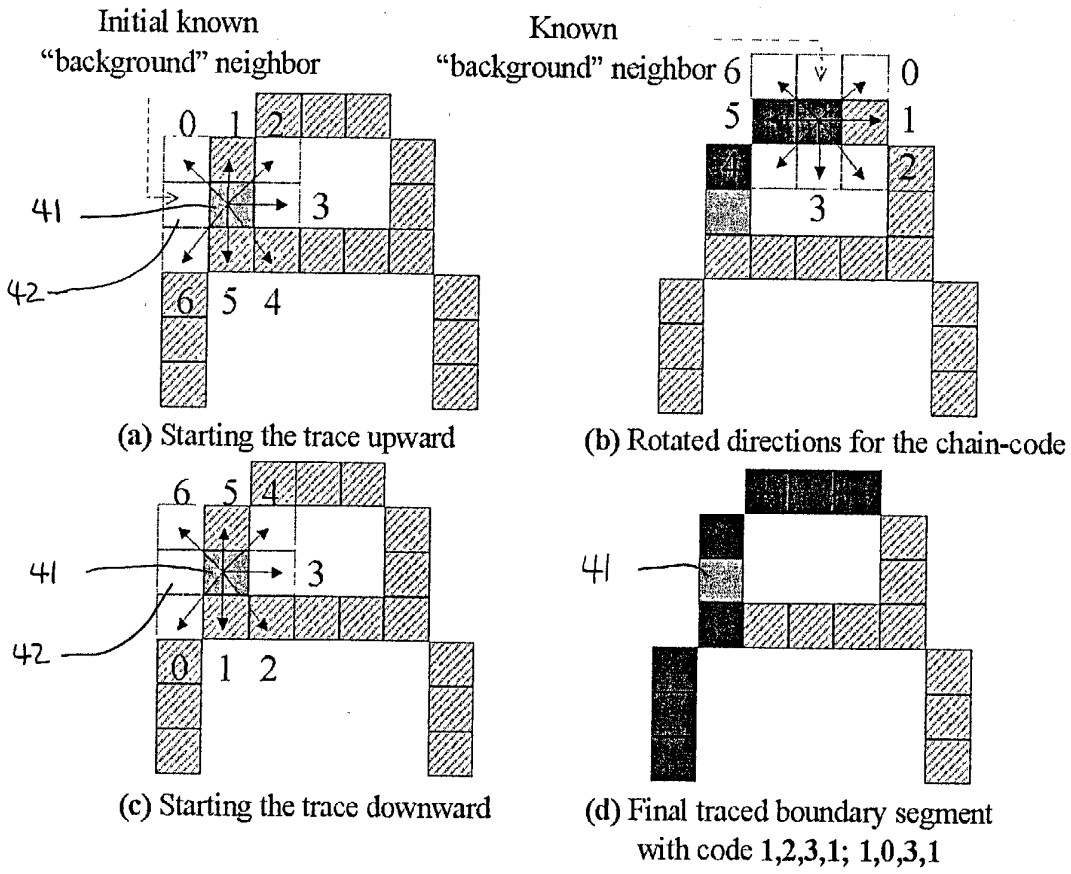

Initial known
"background" neighbor
0
1
2
3
41
42
6
5
4
(a) Starting the trace upward
Known
"background" neighbor
6
0
5
1
4
2
3
(b) Rotated directions for the chain-code
6
5
4
41
3
42
0
1
2
(c) Starting the trace downward
41
(d) Final traced boundary segment
with code 1,2,3,1; 1,0,3,1

FIG. 4

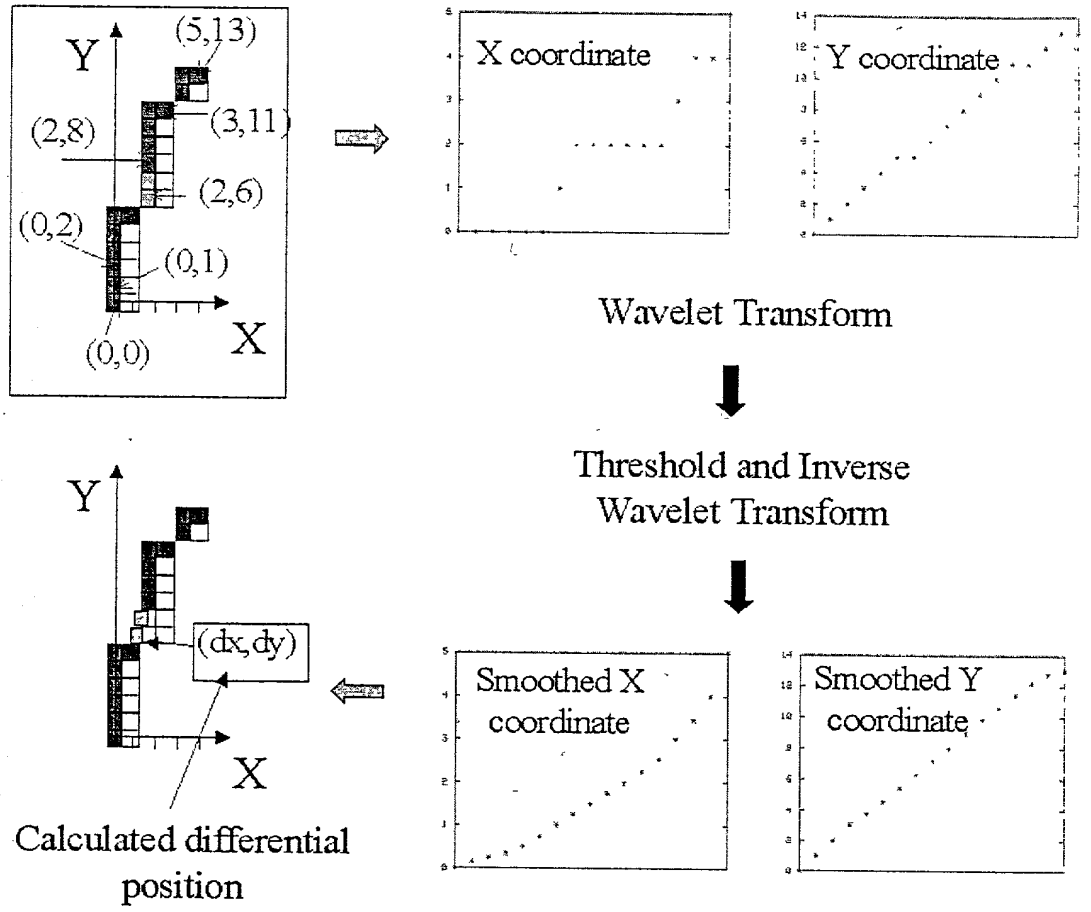

Y
(5,13)
(2,8)
(3,11)
(2,6)
(0,2)
(0,1)
(0,0)
X
X coordinate
Y coordinate
Wavelet Transform
Threshold and Inverse
Wavelet Transform
Smoothed X
coordinate
Smoothed Y
coordinate
Y
(dx,dy)
X
Calculated differential
position

FIG. 5

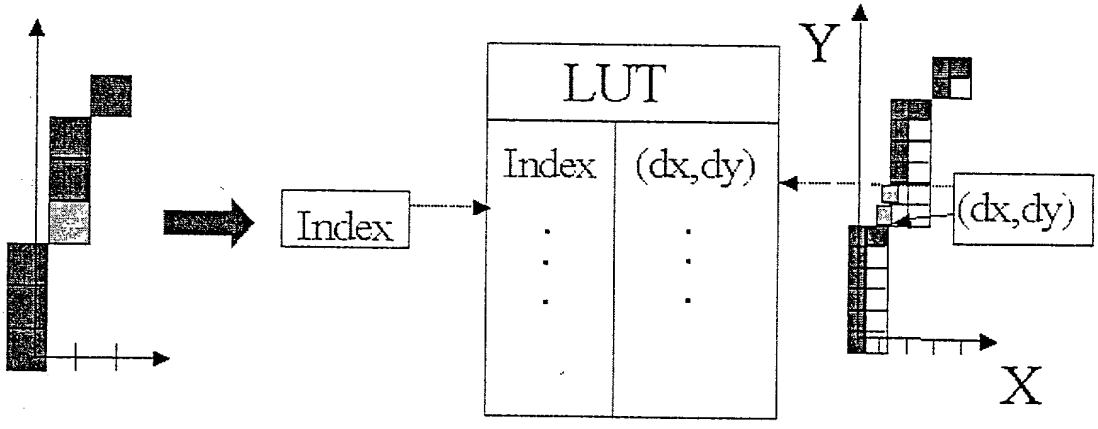
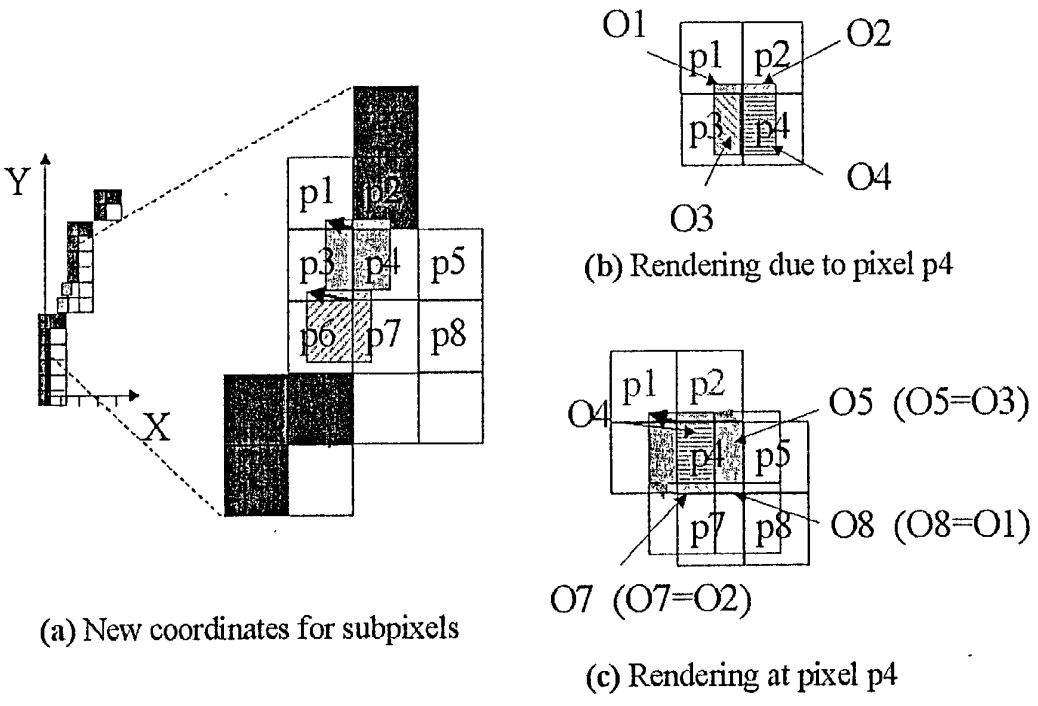

Index
LUT
Index
(dx,dy)
Y
(dx,dy)
X

FIG. 6



(a) New coordinates for subpixels

(b) Rendering due to pixel p4

(c) Rendering at pixel p4

FIG. 7

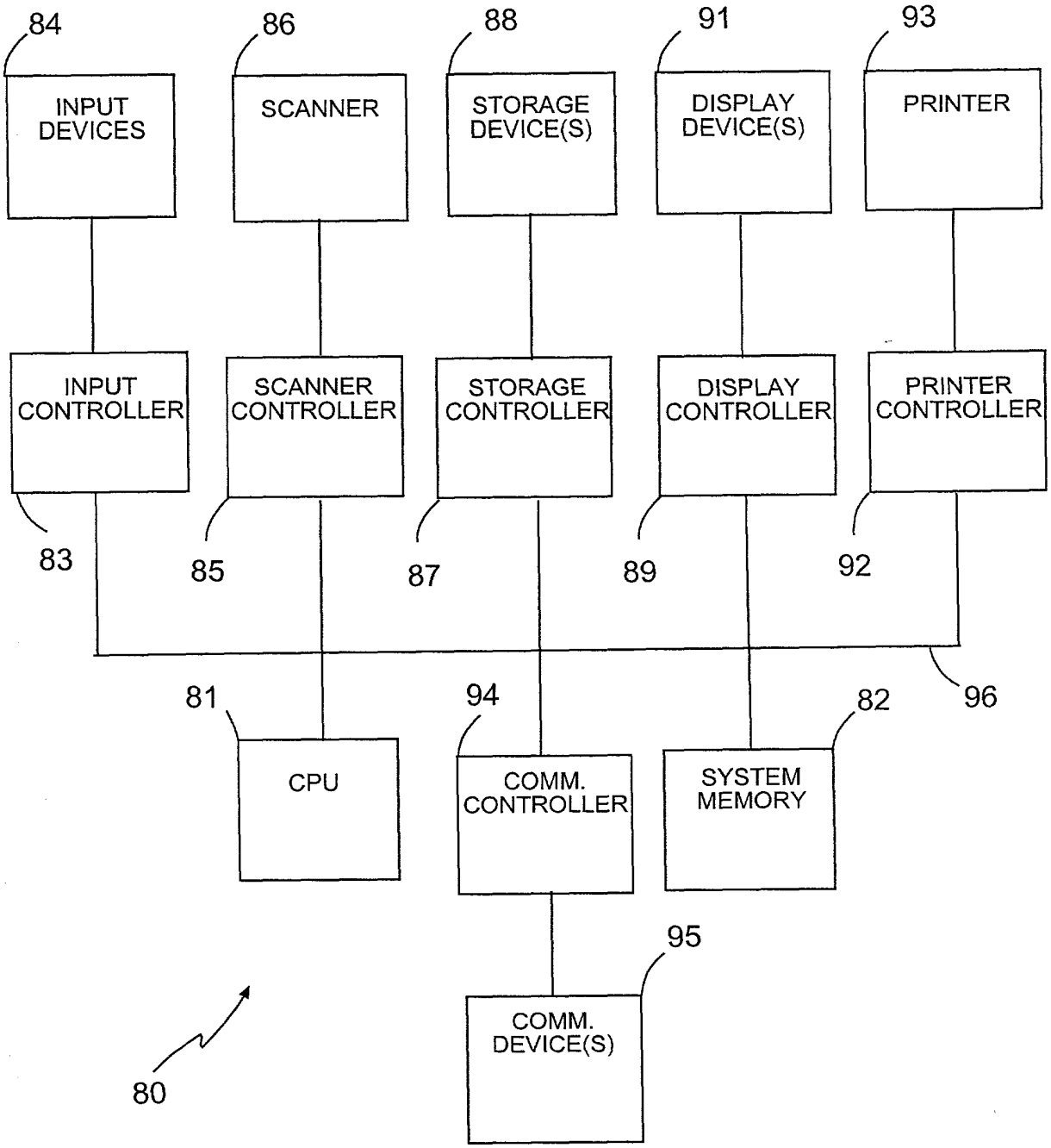

84
INPUT DEVICES
86
SCANNER
88
STORAGE DEVICE(S)
91
DISPLAY DEVICE(S)
93
PRINTER
INPUT CONTROLLER
SCANNER CONTROLLER
STORAGE CONTROLLER
DISPLAY CONTROLLER
PRINTER CONTROLLER
83
85
87
89
92
96
81
CPU
94
COMM. CONTROLLER
82
SYSTEM MEMORY
95
COMM. DEVICE(S)
80

FIG. 8